

AUCTION ALGORITHM FOR WEAPONS/TARGETS PAIRING APPLICATION

Zbigniew Bogdanowicz*, Norman Coleman
US Army Armament R&D Center
Picatinny Arsenal, New Jersey 07806, USA

ABSTRACT

In this paper we propose and study the performance of optimizing weapons/targets pairing based on an *auction algorithm*. The weapons/targets pairing problem can be considered as an assignment optimization problem in mathematics. Hence, there are number of optimal algorithms that can solve it. We show that for practical weapons/targets pairing a well-known in the literature auction algorithm should be considered as a preferred choice.

1. INTRODUCTION

An optimal assignment of weapons to targets is a challenging optimization problem. With the advancement of software technology and algorithms it becomes more realistic and practical every day for modern battlefields. Furthermore, the outcome of today's modern battles may strongly depend on the intelligent usage of all available weapons maximizing their effectiveness.

In general, the pairing of weapons to targets considers many input types and the strategy of optimization might vary considerably. Hence, designing a single optimization algorithm for generic input is a hard problem. Weapons/targets pairing, however, can be reduced to an assignment optimization problem, which is well known and studied in mathematics (Bertsekas, 1992; Bertsekas, 1990; Castanon, 1993; Galil, 1986; Hopcroft and Karp, 1973; Micali and Vazirani, 1980). So, in this paper we consider a two-step approach to the weapons/targets pairing problem. In the first step, a preprocessing algorithm converts all the input information into a single benefit matrix A , where each a_{ij} in A represents a benefit of assigning row i to column j . In the second step, an optimization algorithm assigns rows to columns in A in such a way that the total benefit is maximized. We focus in this paper on the second step of the above approach.

2. MATHEMATICAL FORMULATION

Let a_{ij} be a value of assigning weapon i into target j . Let $x_{ij}=1$ indicate that weapon i is assigned into target j ,

otherwise $x_{ij}=0$. In general, the number of weapons n_1 does not have to equal to the number of targets n_2 . So, we have the following weapons/targets pairing mathematical formulation.

$$\max \sum_{i,j} a_{ij} \cdot x_{ij} \quad (1)$$

subject to

$$\sum_i x_{ij} \leq 1 \quad (2)$$

$$\sum_j x_{ij} \leq 1 \quad (3)$$

The input to auction algorithm is matrix $A=[a_{ij}]_{n_1 n_2}$ where a_{ij} represents a benefit of assigning row i to column j (i.e., benefit of assigning weapon i to target j).

3. PERFORMANCE OF AUCTION ALGORITHM

If $n_1 \neq n_2$ then the input A to an auction algorithm can be easily translated into A' with $n_1=n_2=n$ by appending either rows with 0 entries (corresponding to phantom weapons) or columns with 0 entries (corresponding to phantom targets). Hence without loss of generality consider A as a symmetric input to an auction algorithm for analysis of its performance.

Let ε be a minimum increase of bid cost for a target if an auction algorithm assigns a weapon to that target in its iteration. Assuming integral benefit matrix A (i.e., all a_{ij} being integers), the auction algorithm guarantees that the feasible result of optimization is optimal if $\varepsilon < 1/n$ (Bertsekas, 1992).

Consider now the worst-case running time complexity of accomplishing optimal assignment. The assignment problem can be modeled with a bipartite graph $G=(V,E)$, where the number of vertices $|V(G)|=2n$ and the number of edges $|E(G)|=m$ in G . Let $C=\max_{(i,j) \in E(G)} a_{ij}$. The total number of iterations in which a target receives a bid is no more than C/ε . In addition, an auction algorithm can be implemented in such a way that its iteration involves a bid by a single weapon. So, the total number of iterations is no more than n times C/ε , and

Report Documentation Page				Form Approved OMB No. 0704-0188	
Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.					
1. REPORT DATE 00 DEC 2004		2. REPORT TYPE N/A		3. DATES COVERED -	
4. TITLE AND SUBTITLE Auction Algorithm For Weapons/Targets Pairing Application				5a. CONTRACT NUMBER	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S)				5d. PROJECT NUMBER	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) US Army Armament R&D Center Picatinny Arsenal, New Jersey 07806, USA				8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)				10. SPONSOR/MONITOR'S ACRONYM(S)	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release, distribution unlimited					
13. SUPPLEMENTARY NOTES See also ADM001736, Proceedings for the Army Science Conference (24th) Held on 29 November - 2 December 2005 in Orlando, Florida.					
14. ABSTRACT					
15. SUBJECT TERMS					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT UU	18. NUMBER OF PAGES 2	19a. NAME OF RESPONSIBLE PERSON
a. REPORT unclassified	b. ABSTRACT unclassified	c. THIS PAGE unclassified			

since every bid requires $O(n)$ operations, the worst running time of the algorithm is

$$O(n^2 C/\varepsilon) \quad (4)$$

If every benefit $a_{ij} \in A$ is an integer (which we can accomplish in most cases by scaling up every $a_{ij} \in A$ by appropriately large integer), and $\varepsilon < 1/n$, then according to (Bertsekas, 1992) the worst time complexity is

$$O(nm \log(nC)) \quad (5)$$

4. AUCTION ALGORITHM AS A PREFERRED CHOICE

To solve an assignment optimization problem focused on weapons and targets we might first consider if it makes sense to use an exact optimization algorithm vs. an approximate heuristic. Since the number of weapons and targets in realistic battlefield scenarios should run up to tens in most cases, we estimate that an exact optimization algorithm should perform time-efficiently (i.e., in order of seconds at the worst). Hence our attention should be focused on finding an exact optimization algorithm for the assignment problem of this range.

There are number of optimal algorithms that can solve it and they are well documented in the literature – *shortest augmented path algorithm*, *interior point based algorithm*, *auction algorithm* to name a few (Adler et al., 1989; Galil, 1986; Hopcroft and Karp, 1973; Karmarkar, 1984; Micali and Vazirani, 1980; Todd and Low, 1992;). We show that for weapons/targets pairing an auction algorithm should be considered as a preferred choice. In addition, we back this up with the performance results based on a simple *forward auction algorithm* implementation.

For the above input size a well-implemented auction algorithm should run in order of seconds, as it has been verified through the computations on 40 test cases. Furthermore, the bidding and assignment phases of the auction algorithm are highly parallelizable, which makes the auction algorithm scalable. That is, the bidding and the assignment can be carried out for all weapons and targets simultaneously, which could extend the range of input to hundreds of weapons and targets, and beyond.

The nature of weapons/targets pairing should allow a cost scaling, which could produce matrix A with all integral benefits a_{ij} . This in turn, could further improve the performance of the auction-based algorithms.

For much larger input sizes one could also consider other assignment algorithms. For very large input sizes (i.e.,

thousands of weapons and targets) an interior point algorithm was shown to beat a simply implemented auction algorithm. However, such inputs would be extremely rare for weapons/targets pairing in the real world. In addition, one could still address this class of problems with the parallel implementation of an auction algorithm.

5. COMPUTATIONAL RESULTS

We implemented a simple forward auction algorithm in PC/Linux RedHat 9.0 environment. Our implementation supports input of up to 1000 weapons by 1000 targets. However, in the study conducted here we generated inputs of up to 120 weapons by 120 targets. Based on 40 test cases our implementation of the auction algorithm produced all optimal solutions predominantly in the order of milliseconds. For four test cases with larger inputs our algorithm executed over 10s with the worst-case rate of 0.64s per target. The optimal solution resulted in gain of up to 7% in comparison to the initial ad hoc heuristic-based solution. The average improvement resulted in 2.9%, and the average execution time was 3.9s.

REFERENCES

- Adler, I., Karmarkar, N. and Resende M., 1989: An Implementation of Karmarkar Algorithm for Linear Programming, *Math. Programming*, **44**, 297-335.
- Bertsekas, D., 1992: Linear Network Optimization: Algorithms and Codes, Reading, The MIT Press.
- Bertsekas, D., 1990: The Auction Algorithm for Assignment and Other Network Flow Problems, *Interfaces* **20** 133-149.
- Castanon, D., 1993: Reverse Auction Algorithms for Assignment Problems, *Algorithms for Network Flows and Matching – American Math. Soc.*, 407-429.
- Galil, Z., 1986: Efficient Algorithms for Finding Maximum Matchings in Graphs. *ACM Computing Surveys*, **18**, 23-38.
- Hopcroft, J. and Karp, R., 1973: An $n^{5/2}$ Algorithm for Maximum Matchings in Bipartite Graphs, *SIAM J. Computing* **2**, 225-231.
- Karmarkar, N., 1984: A New Polynomial-time Algorithm Linear Programming, *Combinatorica* **4**, 373-395.
- Micali, S. and Vazirani V., 1980: An $O(\sqrt{|V|} |E|)$ Algorithm for Finding Maximum Matchings in General Graphs, *Proc. 21st. Symp. Foundations of Computing*, 17-27.
- Todd M. and Low, A., 1992: Complexity Interior Point Algorithm for Linear Programming, *SIAM Journal of Optimization* **2**, 198-209.